멀티 애플리케이션 스마트 카드를 위한 스마트 카드 애플릿 관리 시스템

은나래, 조동섭
이화여자대학교 컴퓨터학과
e-mail : naraeun@ewhain.net, dscho@ewha.ac.kr

Smart Card Applet Management System for Multi-Application Smart Card

Narae Eun, Dong-Sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

여러 개의 프로그램을 저장할 수 있고 카드를 카드 소유자에게 전달한 후에도 필요한 프로그램을 다운로드 받을 수 있고 불필요한 프로그램은 제거할 수 있는 멀티 애플리케이션 스마트 카드에 대해 연구한다. 그리고 멀티 애플리케이션 스마트 카드의 특징에 대해 알아보고 이를 통해 멀티 애플리케이션 스마트 카드에 저장되는 프로그램인 애플릿에 대한 관리 방안에 대해 제안한다.

1. 서론

최근 유비쿼터스 컴퓨팅에 관심이 높아짐에 따라 이동성을 가지면서 안전한 방법으로 정보를 저장할 수 있는 스마트 카드가 중요한 위치를 차지하게 되었다. 기존 스마트 카드는 전자 인증 기능만을 가지거나 또는 간단한 개인정보 정도만 저장할 수 있었다. 한 개의 카드에는 한 개의 애플리케이션만 저장할 수 있었다. 그러나 저장 공간이 늘어나고 처리 능력이 증가함에 따라 스마트 카드에는 정보만 저장할 뿐만 아니라 여러 프로그램을 저장할 수 있고 저장한 프로그램을 수행할 수 있게 되었다.[1] 또한 한 개의 카드로 여러 가지의 기능을 사용할 수 있는 멀티 애플리케이션 스마트 카드에 대한 연구가 많이 진행되고 있다. 그리고 Java Card, MULTOS, Windows for Smart Cards 같이 멀티 애플리케이션 스마트 카드를 지원하는 플랫폼도 나타나게 되었다.

멀티 애플리케이션 스마트 카드는 카드 소유자에게 카드가 전달되기 이전에 이미 저장되어 있는 프로그램을 가지고 있을 뿐만 아니라 카드 소유자가 원하는 프로그램을 스마트 카드에 다운로드를 통해 전달 받아 저장하여 프로그램을 가질 수 있다. 게다가 카드에 저장되어 있는 프로그램 중 더 이상 필요하지 않는 프로그램은 제거할 수 있다. 이를 통하여 여러 개의 카드를 가지지 않고도 한 개의 스마트 카드로 다양한 프로그램을 수행시켜 여러 가지 기능을 사용할 수 있게 되었다.

스마트 카드가 카드 사용자에게 전달 된 후 애플리케이션을 설치하고 제거할 수 있기 때문에 여기에 대한 관리가 필요하다. 그리고 여러 애플리케이션이 함께 공존하기 때문에 애플리케이션간 정보를 가지고 있고, 이런 애플리케이션에 대해 관리해주는 시스템이 필요하다. 본 연구에서는 스마트 카드 애플리케이션에 대해 관리해주는 애플리케이션 관리 방법에 대해 제안하고자 한다.

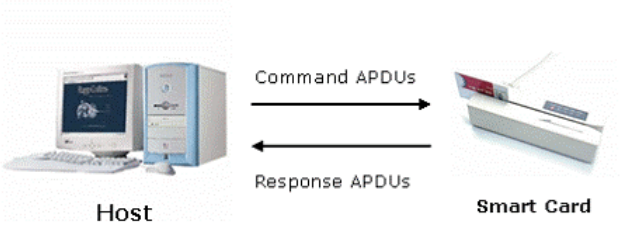
2 장에서는 관련 연구로 스마트 카드와 Open Card Framework 에 대해 설명하고 3 장에서는 멀티 애플리케이션 스마트 카드에서 필요한 관리 방법에 대해 설명하고 4 장에서는 결론을 기술한다.

2. 관련 연구

2.1 Smart Card

스마트 카드 애플리케이션은 카드 리더, 호스트에서 실행되는 오프-카드 애플리케이션과 스마트 카드

칩에서 실행되는 온-카드 애플리케이션으로 나눌 수 있다. 오프-카드 애플리케이션과 온-카드 애플리케이션 사이에는 데이터가 전송이 되는데 이 때 전송되는 데이터 패킷을 APDU(Application Protocol Data Unit)라고 부른다.[3] APDU 는 호스트에서 스마트 카드에 명령을 전달하는 command APDU, 스마트 카드에서 호스트로 전송하는 response APDU 가 있다. [그림 1] 그래서 호스트와 스마트 카드 사이에는 APDU 을 통해 데이터를 주고 받으면서 애플리케이션이 실행이 된다.



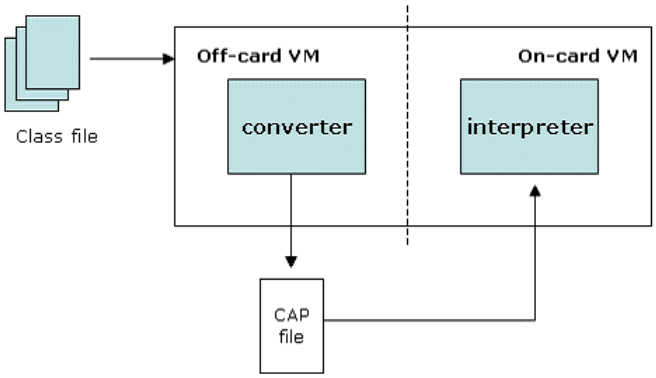
[그림 1] 스마트 카드 통신

스마트 카드가 소비자에게 전달된 후에 프로그램을 다운로드 받을 수 있는 프로그램 코드[4]로는 실제 하드웨어에 맞는 기계 코드를 다운로드 받는 경우와 인터프리터 코드를 다운로드 받는 경우 두 가지로 나뉠 수 있다. 기계 코드를 다운로드 받을 경우에는 이미 컴파일된 코드이기 때문에 카드에 인터프리터가 올라와 있을 필요가 없다. 그래서 빠르게 수행할 수 있다. 그러나 메모리 관리에 대해 신경써야 한다. 인터프리터 코드는 카드에 올라와 있는 인터프리터를 통해 실행되므로 속도는 느리지만 메모리 영역에 대해 고려하지 않아도 되고 하드웨어와 독립적으로 운영할 수 있는 장점을 가지고 있다.

다음 장에서는 다운로드 받을 수 있는 프로그램 코드 중 인터프리터 코드로 분류되는 자바 카드에 대해 알아보겠다.

2.2 Java Card

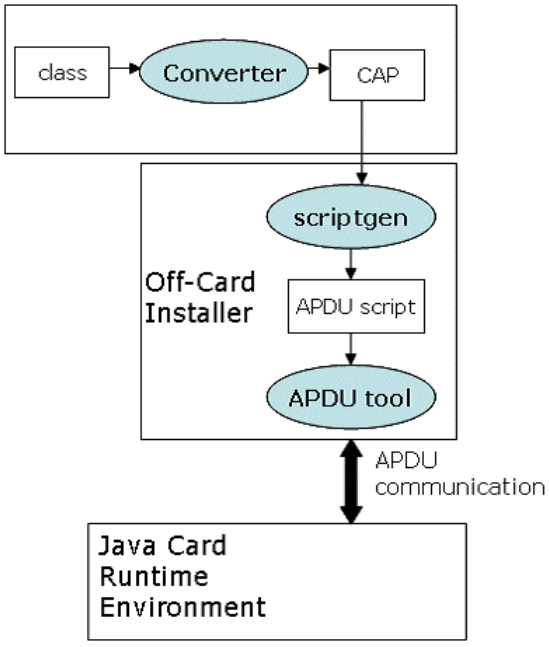
Java Card 는 스마트 카드에서 자바 프로그래밍 언어로 구현된 프로그램을 실행시킬 수 있다. 이 때 자바 프로그래밍 언어로 구현된 프로그램을 애플릿이라고 한다. 자바 카드는 스마트 카드 애플리케이션을 고급 언어를 사용하여 쉽게 구현할 수 있도록 하는 장점[2]을 가진다. 자바 프로그래밍 언어가 가지는 하나의 프로그램으로 어느 하드웨어에서든 자바 가상 머신[그림 2]을 통해 실행될 수 있는 특징을 가진다. 하지만 스마트 카드의 저장 공간과 컴퓨팅 능력의 제약으로 자바 언어가 가지는 모든 특징을 다 가지고 있지는 않다. 예를 들어 동적 클래스 로딩이나 가비지 콜렉션 같은 기능은 지원을 하지 않는다.[5] 그래서 실제로 자바 카드에서 애플릿을 다운로드 받는 방법은 다음 과정을 거친다.



[그림 2] 자바 카드 가상 머신

카드에 애플릿을 로드시키고 카드에서 애플릿을 설치한다. 애플릿 설치로 카드에서 그 프로그램을 실행시킬 수 없고 카드에 애플릿을 설치한 다음에 애플릿에 대해 개인화가 이루어져야 한다. 여기서 개인화란 자바 카드에 설치한 애플릿에 대해 개인 정보가 추가되어 실제로 카드에서 실행할 수 있도록 하는 과정이다. 이 과정은 모두 호스트와 카드와의 통신을 통해서 이루어진다.

자바 카드에 애플리케이션을 설치하고 실행되기 위해서 [그림 3]과 같은 과정을 거친다.[6]



[그림 3] 자바 카드 구조

1. 자바 프로그래밍 언어로 자바 카드 애플리케이션을 구현한다.
2. 자바 카드 애플리케이션을 클래스 파일로 변환한다.
3. 클래스 파일은 Converter 툴의 인풋으로 들어가 CAP 파일을 생성한다. CAP 파일이란 converted applet 으로 자바 클래스를 자바 패키지로 만든 이진 파일이다.

4. CAP 파일은 오프카드 인스톨러인 scriptgen 툴의 인풋으로 들어가 APDU 스크립트 파일을 생성한다.

5. 생성된 스크립트 파일은 apdu 툴을 이용하여 자바 카드와 APDU 명령을 주고 받는다.

3. 스마트 카드 애플릿 관리

3.1 OpenCard Framework

Open Card Framework 는 서로 다른 하드웨어 플랫폼을 가지고 있고 다른 운영체제를 가지고 있어도 제조사가 다른 카드 리더기에서 스마트카드 정보에 접근할 때 같은 애플리케이션을 사용할 수 있도록 해 주는 프레임워크이다. 하드웨어 플랫폼, 운영체제, 제조사에 상관없이 같은 애플리케이션을 사용할 수 있도록 인터페이스를 제공해준다. 제조사에 독립적인 인터페이스를 제공하므로 애플리케이션 개발시간도 줄어들며 카드 리더기 제조사, 카드 운영체제 제공자, 카드 제공자에게도 도움을 준다. Open Card Framework 는 크게 utility class, terminal layer, service layer, security 로 나뉜다. 애플릿 관리를 하는데 사용하는 클래스는 service layer 중 Application Managemet CardService 이다. Application Management CardService 를 이용하여 applet 의 상태를 체크할 수 있고, applet 을 선택, 설치, 제거를 할 수 있다.

3.2 멀티애플리케이션 스마트 카드

멀티 애플리케이션 스마트 카드가 가져야 할 특징은 다음과 같다. 첫째, 여러 프로그램을 공존하기 위해서는 기술적인 표준이 필요하고 프로그램 간 상호 운용성이 보장되어야 한다. 둘째, 애플리케이션과 키를 다운로드하고 제거하는 과정에서 무결성이 보장되어야 한다. 셋째, 전체 과정을 관리하고 제어할 수 있어야 한다.

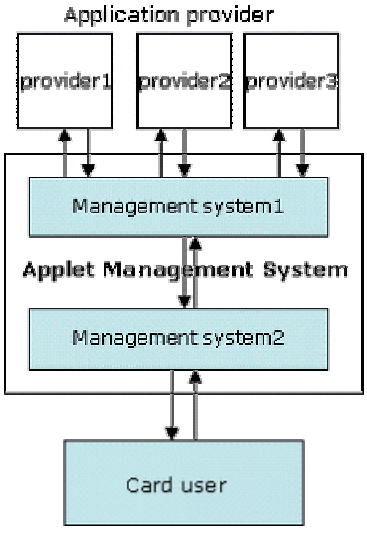
멀티 애플리케이션 스마트 카드에 관련된 참여자는 사용자, 카드 발행자, 애플리케이션 제공자 크게 셋으로 분류할 수 있다. 참여자에게 관리 권한을 어떻게 주느냐에 따라 관리 방법도 달라진다. 모든 권한을 전적으로 카드 발행자에게 주게 되면 애플리케이션 제공자는 카드 발행자와 상의하여 프로그램을 제공한다. 다른 방법으로는 카드 사용자가 모든 권한을 갖도록 하는 방법도 있다. 카드 생산자는 단지 빈 카드만을 제공하며 빈 카드를 산 사용자는 자신이 필요한 프로그램의 애플리케이션 제공자에게 프로그램을 구입하여 설치할 수 있다.[7] 본 연구에서는 카드 생산자가 모든 권리를 가지고 관리하도록 하고 애플리케이션 제작자는 카드 생산자에게 애플리케이션을 제공하는 방법을 사용한다.

같은 카드 내에서 여러 개의 애플리케이션 제공자가 제공하는 애플리케이션을 내장하고 있기 때문에 애플리케이션 간 협력이 중요하다. 애플리케이션을 효율적으로 카드에 설치하고 제거하는 매커니즘이 필요하다.

한정되어 있는 메모리에 여러 프로그램이 내장되기 때문에 애플리케이션간 공유가 필요하다. 애플리케이션간 객체 공유를 위해서는 공유에 관계되는 애플리케이션끼리 보안에 대한 정책이 필요하다. 이것은 애플리케이션 제공자끼리만의 약속으로 이루어지는 것이 아니라 카드 전체에 대해 보안에 대한 정책이 이루어져야 한다. [8] 그러므로 관리 시스템에서 이에 대한 보안 정책을 수립하고 실제 스마트 카드에서 공유가 일어날 때 아무런 문제없이 실행되어야 한다.

3.3 애플릿 관리 시스템

애플릿 관리 시스템은 크게 스마트 카드에 내장되어 있는 애플릿에 대한 관리, 애플리케이션 제공자가 제공하는 애플리케이션에 대한 관리로 나눌 수 있다. [그림 4]에서 애플릿 관리 시스템 구조를 볼 수 있다. 여기서 애플리케이션 제공자가 제공하는 애플리케이션에 대한 관리는 system1 에서, 스마트 카드에 내장되어 있는 애플릿에 대한 관리는 system2 에서 이루어진다.



[그림 4] 애플릿 관리 시스템 구조

우선 애플리케이션 제공자가 제공하는 애플리케이션에 대한 관리 시스템에서 이루어져야 하는 관리를 알아보면 다음과 같다. 애플리케이션간 공유문제에 대해 정보 흐름도를 통해 재공유가 이루어지지 않도록 해야 한다.

스마트 카드 내에 내장되어 있는 애플리케이션에 대한 관리 시스템에서는 스마트 카드의 life cycle 동안 스마트 카드에 내장되어 있는 애플리케이션이 어떤 상태인지에 대한 조사하고 오류 발생여부에 대한 파악이 이루어져야 한다. 그리고 애플리케이션의 동적인 설치, 제거에 대한 메커니즘이 존재해야 하며, 애플리케이션의 설치 및 제거가 일어났을 때 무결성 보장이 되어야 한다.

애플리케이션 제공자가 제공하는 애플리케이션 관리와 스마트 카드에 내장되어 있는 애플리케이션 관리를 분리시켜 관리 시스템을 구성했지만 서로 두 개의 시스템은 서로 연결되어 있는 하나의 시스템이다. 서로 정보를 교환하면서 애플릿의 제거, 설치 여부를 조사하여 애플릿에 대한 이용도를 파악할 수 있고, 애플릿에 대한 공유 설정 정보를 통해 애플릿의 업데이트 시 필요한 것만 설치할 수 있도록 할 수 있다.

4. 결론

본 논문에서는 여러 개의 애플리케이션을 내장할 수 있고 카드 사용자에게 전달 된 후에도 애플리케이션을 설치, 제거할 수 있는 멀티 애플리케이션 스마트 카드에 대해 연구하였다. 서로 다른 애플리케이션이 공존하기 때문에 애플리케이션간 협력과 상호 운용성이 중요하다. 이 때문에 관리 시스템에 대한 중요도가 높아진다. 연구에서는 카드 생산자가 모든 권한을 가지고 있도록 하였고 스마트 카드 내에 내장되어 있는 애플리케이션에 대한 관리와 애플리케이션 제공자가 제공하는 애플리케이션에 대한 관리로 두 가지로 나누어서 알아보았다. 이를 통해 카드 사용자와 애플리케이션 제공자의 연결이 쉽게 가능하고, 두 시스템간의 정보를 교환하면서 애플릿 설치, 제거 정보를 통한 애플릿에 대한 이용도 조사를 할 수 있고, 공유 설정 정보를 통한 업데이트 용량을 줄일 수 있는 효과도 얻을 수 있을 것이다.

참고문헌

[1] JAVACARD, Juha-Pekka Ruuskanen
[2] Advanced Control Flow in Java Card Programming, Peng Li, Steve Zdancewic, LCTES'04, 2004.6
[3] Developing Smart Card Applications Using the OpenCard Framework, Mark Burge, ACMSE'04, 2004. 6
[4] Smart Card HandBook 2nd Edition, W. Rankl, W. Effing, JOHN WILEY & SONS, LTD
[5] Smart Card Evolution, Katherine M. Shelfer, J. Drew Procaccino, Communication of the ACM, 2002. 7
[6] Smart Card HandBook 2nd Edition, W. Rankl, W. Effing, JOHN WILEY & SONS, LTD
[7] Which Security Policy for Multiapplication Smart Cards?, Pierre Girad, USENIX Workshop on Smartcard Technology, 1999.5s
[8] Java Card or How to Cope with the New Security Issues Raised by Open Cards?, Pierre Girard, Jean-Louis Lanet, Gemplus Developer Conference'99, 1999.6